UNIVERSITY OF SASKATCHEWAN
DEPARTMENT OF COMPUTATIONAL SCIENCE
CMPT 832/417 MIDTERM EXAM

1.5 hours | Closed Book | November 23, 1993

1. (12 marks)

> Depth first search (DFS) is space efficient but gets hung up on infinite loops, possibly missing many answers. Breadth first search (BFS) is complete in that it eventually finds every answer, but its space complexity makes it impractical for combinatorial search.
>
> Iterative deepening search (IDS) gives us the best of both worlds — completeness with space efficiency.
>
> — Kim Campbell

Give short answers to the following questions.

(a) Give a small (trivial) search problem that illustrates what Kim is trying to say.

(b) Suppose we represent 100 cities as nodes and paths between them as arcs. True or false — it would be possible to perform a depth first search of the graph on a microcomputer, but it would be highly unlikely to be able to perform a breadth first search on the same computer. Justify your answer.

(c) True or false — it takes about half as long to find the first solution to a problem with DFS as with BFS. Justify your answer.

(d) True or false — DFS and BFS take about the same time to find all solutions to a problem. Justify your answer.

(e) True or false — IDS takes about the same time as DFS to find the first answer to a problem. Justify your answer.

(f) When is loop-checking a better strategy than iterative deepening?

2. (12 marks)

In the classical logics, if C is any contradiction, and B is any sentence, the sentence

```
C => B
```

is always true. Thus, given a single contradiction, every sentence and its contrary becomes true.

This is not true in computational logics where resolution is the only inference rule. That is, if a database is inconsistent, it is not possible to derive arbitrary sentences.

In light of these fact, how can resolution based systems be useful as consistency-checkers?

3. (8 marks)
Consider the English language sentences:

```
If the ore is copper it has a green colour.
If the ore is from near Sudbury, it is copper or nickel.
The colour is brown.
```

(a) Show how to represent this as a set of sentences of propositional logic.

(b) Show how to prove the sentence "if the ore is from Sudbury, then it is nickel", using

i. Resolution (any resolution strategy will do)

ii. Prolog-like proof tree with negated ancestor resolution.

(Hint: there is a piece of information implicit in the representation needed to make the proof go through.)

4. (18 marks)
Consider the following Theorist program:

```
fact penguin(willy).
fact penguin(chilly).
fact antarctic(X) and bird(X) => penguin(X).
fact bird(willy).
fact bird(tweety).
default bf(X): bird(X) => fly(X).
default pnf(X): penguin(X) => not fly(X).
fact fly(chilly).
```

What answers does Theorist give to the following queries. Try to give all answers.

(a) explain not fly(chilly).

(b) explain fly(chilly).

(c) explain fly(willy).

(d) explain not fly(willy).

(e) explain not antarctic(tweety).

(f) explain not antarctic(chilly).

5. (12 marks)
Suppose you are implementing a planner for automatic program consruction. You have massive reusuable software modules (numbered m_1, m_2, etc.) and each can be characterized in terms of its postconditions (what it does) and its preconditions (what it needs to be true before it can execute). Denote the preconditions of module 1 by $precond_1$ and the postconditions of module 2 by $postcond_2$.

Give a simple Theorist or ATMS description of these modules that, given a set of postconditions (specifications of a problem to be solved), returns a possible set of modules needed to build the program. Show how your system words by identifying the facts, defaults, observations needed, and show how to pose a query that returns the desired information.

6. (1 mark) (Commonsense reasoning)
Which is better, to work 100 hours for $1.00/hour, or 1 hour for $100.00/hour.